

(11)Publication number : 05-094470
(43)Date of publication of application : 16.04.1993

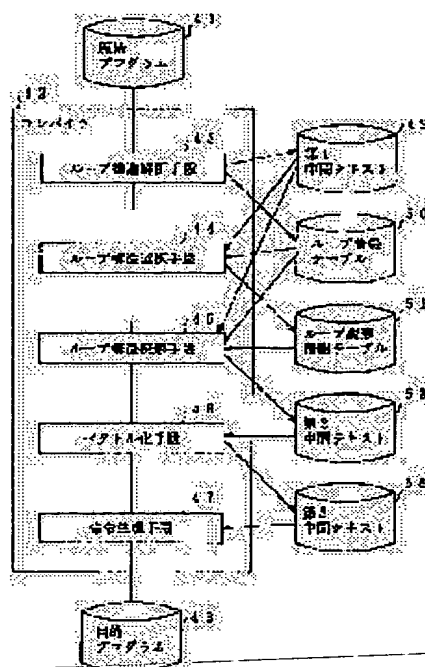
G06F 15/347
G06F 9/45

(71)Applicant : NEC CORP

(72)Inventor : YOKOYA YUJI

(57)Abstract:

CONSTITUTION: As for the vector computer having a vector command of the recurrence formula in an $x(i)=f(x(i-1))$ form, in a compiler 42 which inputs an original program 41 described in a high-level language, and prepares an objective program 48, a loop structure selecting means 44 selects the loop including the substitution sentence in the $x(i)=f(x(i-n))$ form while (n) is the multiple of an increment(incr), among the loop structures of an initial value(init), terminal value(term) and increment(incr). A loop structure transforming means 45 transforms the selected loop structure into a dual loop structure constituted of an outside loop of an initial value (0), terminal value(n/incr), increment (1), and loop control variable(i), and an inside loop of an initial value (init + (i)), terminal value(term) and increment(n). A vectorizing means 46 vectorizes the inside loop of the trans



[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

THIS PAGE BLANK (USPTO)

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-94470

(43) 公開日 平成5年(1993)4月16日

(51) Int.Cl.⁵

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/347
9/45

G 6798-5L

9292-5B

G 0 6 F 9/44

3 2 2 G

審査請求 未請求 請求項の数 2 (全 9 頁)

(21) 出願番号 特願平3-278322

(22) 出願日 平成3年(1991)9月30日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 横谷 雄司

東京都港区芝五丁目7番1号 日本電気株式会社内

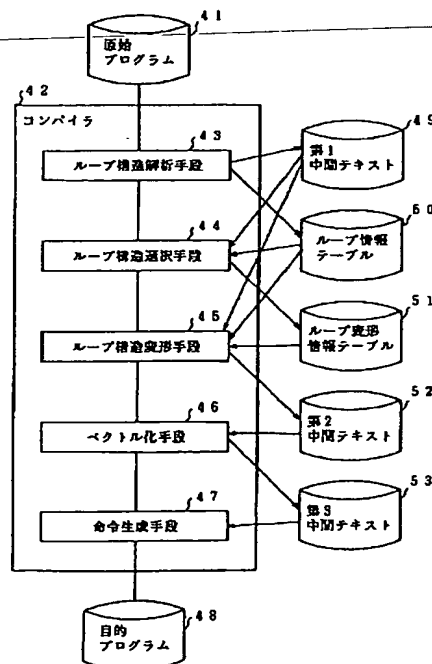
(74) 代理人 弁理士 境 廣巳

(54) 【発明の名称】 ベクトル化方式

(57) 【要約】

【目的】 $x(i) = f(x(i-n))$ の形の式を含むループをベクトル化して、ベクトル計算機上で高速に実行できる目的プログラムを生成する。

【構成】 $x(i) = f(x(i-1))$ の形をした漸化式のベクトル命令をもつベクトル計算機に対して、高級言語で記述された原始プログラム41を入力し目的プログラム48を生成するコンパイラ42において、ループ構造選択手段44は初期値 $init$ 、終値 $term$ 、増分値 $incr$ のループ構造のうち $x(i) = f(x(i-n))$ の形の代入文を含み、しかも n が $incr$ の倍数であるようなループを選択し、ループ構造変形手段45は選択されたループ構造を初期値0、終値 n/i 、増分値1、ループ制御変数 i の外側ループと初期値 $init+i$ 、終値 $term$ 、増分値 n の内側ループとからなる2重ループ構造に変形し、ベクトル化手段46は変形された2重ループの内側ループをベクトル化する。



【特許請求の範囲】

【請求項1】 高級言語で書かれた原始プログラムを入力し、ベクトル計算機に対する目的プログラムを生成するコンパイラにおいて、

原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値 $init$ 、終値 $term$ 、増分値 $incr$ の情報を格納したループ情報テーブルを生成するループ構造解析手段と、

該ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、ループ構造内に現れる配列の定義・参照関係を解析し、配列の定義参照関係の情報を格納した定義参照テーブルを生成する定義参照関係解析手段と、

該定義参照関係解析手段で求められた定義参照テーブルを入力し、前記ループ構造中に後方依存関係のある配列があった場合、該ループ構造中で該配列の各要素が参照される繰り返し回と定義される繰り返し回の差 n を求め、その差 n の値がループ構造内で一定であるループ構造を選択するとともに、 n を格納したループ変形情報

テーブルを生成するループ構造選択手段と、
該ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成されたループ変形情報テーブルとを入力し、前記ループ構造を、初期値 $init$ 、終値 $term$ 、増分値 $incr \times n$ の指標変数 i をもつ外側ループと元のループ構造の i から $min(i + (n-1) \times incr, term)$ の繰り返しを実行する内側ループとからなる2重ループ構造に変形した第2中間テキストを生成するループ構造変形手段と、

該ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形するベクトル化手段と、

該ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する命令生成手段とを含むことを特徴とするベクトル化方式。

【請求項2】 高級言語で書かれた原始プログラムを入力し、 $x(i) = f(x(i-1))$ の形式の漸化式を処理するベクトル命令をもつベクトル計算機に対する目的プログラムを生成するコンパイラにおいて、

原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値 $init$ 、終値 $term$ 、増分値 $incr$ の情報を格納したループ情報テーブルを生成するループ構造解析手段と、

該ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、 $x(i) = f(x(i-n))$ の形式の式を含みしかも n

が前記増分値 $incr$ の倍数となっているループ構造を選択するとともに n を格納したループ変形情報テーブルを生成するループ構造選択手段と、

該ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成されたループ変形情報テーブルとを入力し、前記ループ構造を、初期値0、終値 $n / incr$ 、増分値1で指標変数 i をもつ外側ループと初期値 $init + i$ 、終値 $term$ 、増分値 n の内側ループとからなる2重ループ構造に変形した第2中間テキストを生成するループ構造変形手段と、

該ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形するベクトル化手段と、

該ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する命令生成手段とを含むことを特徴とするベクトル化方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、高級言語で書かれた原始プログラムを入力してベクトル計算機に対する目的プログラムを生成するコンパイラに関し、特に、ループ構造を高速に実行するようなベクトル命令列を生成するベクトル化方式に関する。

【0002】

【従来の技術】 ベクトル演算命令をもつ計算機（ベクトル計算機）においては、複数の規則的に並んだデータ列（ベクトルデータ）間の演算をベクトル命令により一度に高速に実行できる。そこでFORTRAN言語のような高級言語で記述された原始プログラムを翻訳してベクトル計算機に対する目的プログラムを生成するコンパイラにおいては、原始プログラムを解析して可能なかぎりベクトル命令を用いて処理するような命令列を生成する。これをベクトル化と呼ぶ。ベクトル化は通常ループ構造に対して行われる。

【0003】 一般に、或るループ構造がベクトル化可能かどうかは、そのループ構造中に現れる配列の定義参照関係がベクトル化した場合も保存されるかどうかにより決まる。そして、次の4つの条件のうちのどれかを満たしていれば、定義参照関係がベクトル化に適合することが知られている。

【0004】 ①その配列がループ中のどこでも定義されていない。

②その配列の或る要素に対する定義参照はすべてループの同一の繰り返し回に閉じている。

③その配列の或る要素がループの異なる繰り返しにおいてそれぞれ定義されるか定義かつ参照される場合、その定義あるいは参照の原始プログラム上での出現順序とル

3

ープの繰り返しの進行における出現順序とが一致している。

④その配列の或る要素がループ中で定義されているとき、定義されている配列のループの他の場所で定義または参照されている同一配列の要素との間に重なりがない。

【0005】なお、条件①から③を満たさないような配列要素の定義参照関係を後方依存関係があるという。

【0006】そこで、従来は、ループ構造がベクトル化可能かどうかを上記4つの条件に照らして判断し、可能と判断したループ構造についてベクトル化を行っていた。

【0007】

【発明が解決しようとする課題】したがって、たとえば、図2に示すFORTRAN言語で記述されたループでは、

- ・配列A、Bとも定義されているため条件①は不成立。
- ・配列要素B(n)は、図2の代入文aでn回目の繰り返しのとき定義され、図2の代入文bでn-10回目の繰り返しのとき参照されているため、条件②は不成立。
- ・配列要素B(n)が、n回目の繰り返しのとき図2の代入文aで参照され、n-10回目の繰り返しのとき図2の代入文bで定義されるので定義参照の原始プログラム上での出現順序とループの繰り返しの進行における出現順序とが逆になっており、条件③は不成立。
- ・配列Bの定義される要素は11から110番目、参照される要素は1から100番目である。したがって、11から100番目までの要素が定義参照されるため、条件④は不成立。

となり、条件①から④のすべてが成り立たないため、従来技術では図2のようなループはベクトル化できないという問題点があった。

【0008】また、たとえば、図5に示すFORTRAN言語で記述されたループでも、配列Aの第1番目の要素は、図5の代入文cで、第1回目の繰り返して定義され第i+1回目の繰り返して参照されているため、条件①から④のどれも満たされていないので原則としてベクトル化はできない。

【0009】但し、この図5の例のように配列の各要素の値がその配列のひとつ前の要素の値に依存するような漸化式は科学技術計算においてはよく現れるため、ベクトル計算機のなかには漸化式をベクトル処理できるような特別の命令を備えたものが多い。したがって、このような漸化式のベクトル命令を備えたベクトル計算機用の目的プログラムを生成するコンパイラでは、ループ中の漸化式のパターンを認識することにより図5のようなループもベクトル化していた。

【0010】しかしながら、図6に示す代入文dを含むFORTRAN言語で記述されたループでは、配列Aの各要素は、ひとつ前の要素ではなく、図7(a)に示す

4

ように3つ前の要素の値に依存している。したがって、漸化式のパターンには当てはまらないため従来技術では図6のようなループはベクトル化できないという問題点があった。

【0011】そこで本発明の目的は、従来技術ではベクトル化できなかった図2、図6のようなループをベクトル化することができるベクトル化方式を提供することにある。

【0012】即ち、たとえば図2のループは、1から10、11から20、……、91から100の各繰り返しに分割すると、分割されたそれぞれの繰り返の間では配列Bの定義される要素と参照される要素の間に重なりがないため、それらは別個にベクトル化可能である。本発明のベクトル化方式は、このような変形を自動的に行うことにより、従来技術ではベクトル化できなかったループをベクトル化することにより、より実行性能の高い目的プログラムを生成することを可能にするものである。

【0013】また、たとえば図6のループ構造が実際に行う処理は、図7(b)に示すような配列Aの1、4、7、10、2、5、8、11、3、6、9、12の3組の要素列に対する漸化式演算と同等であり、これらはベクトル化可能である。本発明のベクトル化方式は、このような変形を自動的に行うことにより、従来技術ではベクトル化できなかったループをベクトル化することにより、より実行性能の高い目的プログラムを生成することを可能にするものである。

【0014】

【課題を解決するための手段】図2のようなループのベクトル化を可能にする本発明のベクトル化方式は、高級言語で書かれた原始プログラムを入力し、ベクトル計算機に対する目的プログラムを生成するコンパイラにおいて、原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値init、終値term、増分値incrの情報を格納したループ情報テーブルを生成するループ構造解析手段と、該ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、ループ構造内に現れる配列の定義・参照関係を解析し、配列の定義参照関係の情報を格納した定義参照テーブルを生成する定義参照関係解析手段と、該定義参照関係解析手段で求められた定義参照テーブルを入力し、前記ループ構造中に後方依存関係のある配列があった場合、該ループ構造中で該配列の各要素が参照される繰り返し回と定義される繰り返し回の差nを求め、その差nの値がループ構造内で一定であるループ構造を選択するとともに、nを格納したループ変形情報テーブルを生成するループ構造選択手段と、該ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成された

5

ループ変形情報テーブルとを入力し、前記ループ構造を、初期値 $init$ 、終値 $term$ 、増分値 $incr \times n$ の指標変数 i をもつ外側ループと元のループ構造の i から $\min(i + (n-1) \times incr, term)$ の繰り返しを実行する内側ループとからなる2重ループ構造に変形した第2中間テキストを生成するループ構造変形手段と、該ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形するベクトル化手段と、該ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する命令生成手段とを備えている。

【0015】また、図6のようなループのベクトル化を可能にする本発明のベクトル化方式は、高級言語で書かれた原始プログラムを入力し、 $x(i) = f(x(i-1))$ の形式の漸化式を処理するベクトル命令をもつベクトル計算機に対する目的プログラムを生成するコンパイラにおいて、原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値 $init$ 、終値 $term$ 、増分値 $incr$ の情報を格納したループ情報テーブルを生成するループ構造解析手段と、該ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、 $x(i) = f(x(i-n))$ の形式の式を含みしかも n が前記増分値 $incr$ の倍数となっているループ構造を選択するとともに n を格納したループ変形情報テーブルを生成するループ構造選択手段と、該ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成されたループ変形情報テーブルとを入力し、該ループ構造を、初期値 0 、終値 $n/incr$ 、増分値 1 で指標変数 i をもつ外側ループと初期値 $init+i$ 、終値 $term$ 、増分値 n の内側ループとからなる2重ループ構造に変形した第2中間テキストを生成するループ構造変形手段と、該ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形するベクトル化手段と、該ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する命令生成手段とを備えている。

【0016】

【作用】図2のようなループのベクトル化を可能にする本発明のベクトル化方式においては、高級言語で書かれた原始プログラムを入力し、ベクトル計算機に対する目的プログラムを生成する際、ループ構造解析手段が、原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値 $init$ 、終値 $term$ 、増分値 $incr$ の情

6

報を格納したループ情報テーブルを生成し、定義参照関係解析手段が、ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、ループ構造内に現れる配列の定義・参照関係を解析し、配列の定義参照関係の情報を格納した定義参照テーブルを生成し、ループ構造選択手段が、定義参照関係解析手段で求められた定義参照テーブルを入力し、前記ループ構造中に後方依存関係のある配列があった場合、該ループ構造中で該配列の各要素が参照される繰り返し回と定義される繰り返し回の差 n を求め、その差 n の値がループ構造内で一定であるループ構造を選択するとともに、 n を格納したループ変形情報テーブルを生成し、ループ構造変形手段が、ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成されたループ変形情報テーブルとを入力し、前記ループ構造を、初期値 $init$ 、終値 $term$ 、増分値 $incr \times n$ の指標変数 i をもつ外側ループと元のループ構造の i から $\min(i + (n-1) \times incr, term)$ の繰り返しを実行する内側ループとからなる2重ループ構造に変形した第2中間テキストを生成し、ベクトル化手段が、ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形し、命令生成手段が、ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する。

【0017】また、図6のようなループのベクトル化を可能にする本発明のベクトル化方式においては、高級言語で書かれた原始プログラムを入力し、 $x(i) = f(x(i-1))$ の形式の漸化式を処理するベクトル命令をもつベクトル計算機に対する目的プログラムを生成する際、ループ構造解析手段が、原始プログラム中のループ構造を認識し、該ループ構造に対する第1中間テキストを生成するとともに、該ループの初期値 $init$ 、終値 $term$ 、増分値 $incr$ の情報を格納したループ情報テーブルを生成し、ループ構造選択手段が、ループ構造解析手段で求められた前記ループ構造の第1中間テキストおよびループ情報テーブルを入力し、 $x(i) = f(x(i-n))$ の形式の式を含みしかも n が前記増分値 $incr$ の倍数となっているループ構造を選択するとともに n を格納したループ変形情報テーブルを生成し、ループ構造変形手段が、ループ構造選択手段で選択されたループ構造について前記ループ構造解析手段で生成された第1中間テキストおよびループ情報テーブルと、前記ループ構造選択手段で生成されたループ変形情報テーブルとを入力し、前記ループ構造を、初期値 0 、終値 $n/incr$ 、増分値 1 で指標変数 i をもつ外側ループと初期値 $init+i$ 、終値 $term$ 、増分値 n の内側ループとからなる2重ループ構造に変形した第2中

7

間テキストを生成し、ベクトル化手段が、ループ構造変形手段により生成された第2中間テキストを入力し、2重ループの内側ループ構造をベクトル処理を行う第3中間テキストに変形し、命令生成手段が、ベクトル化手段で生成された第3中間テキストを入力し一連のベクトル命令およびスカラ命令列を生成する。

【0018】

【実施例】次に本発明の実施例について図面を参照しながら説明する。

【0019】図1を参照すると、本発明のベクトル化方式の一実施例を適用したコンパイラ2は、ループ構造解析手段3と、定義参照関係解析手段4と、ループ構造選択手段5と、ループ構造変形手段6と、ベクトル化手段7と、命令生成手段8とから構成され、高級言語で書かれた原始プログラム1を入力し、ベクトル計算機に対する目的プログラム9を生成する。

【0020】なお、図1において、10、14、15、11、12、13はそれぞれコンパイラ2が処理の過程で生成する第1、第2、第3中間テキスト、ループ情報テーブル、定義参照テーブル、ループ変形情報テーブルをあらわす。

【0021】以下、本実施例の各部の機能および動作を説明する。

【0022】コンパイラ2は原始プログラム1を入力し、目的プログラム9を生成する。

【0023】このとき、ループ構造解析手段3は原始プログラム1中に現れるループ構造を認識し、そのループ構造に対する第1中間テキスト10を生成するとともに、そのループ構造の初期値init、終値term、増分値incrを格納したループ情報テーブル11を生成する。

【0024】つぎに定義参照関係解析手段4は、ループ構造解析手段3で生成された第1中間テキスト10およびループ情報テーブル11を入力し、そのループ構造中に現れるすべての配列の定義参照関係を解析し、各配列の原始プログラム中で定義参照されている位置、定義参照される繰り返し回の情報を求め、定義参照テーブル12に格納する。

【0025】つぎにループ構造選択手段5は、定義参照関係解析手段4で生成された定義参照テーブル12を入力し、ループ構造中に後方依存関係をもつ配列の定義参照があるかどうか調べる。もし後方依存関係が或る配列が存在したなら、その配列の各要素が定義される繰り返し回と参照される繰り返し回の差をとり、それをnとする。そして、もしnがそのループ構造内で値が一定ならば、そのループ構造を変形可能としてループ変形情報テーブル13にnを格納してループ構造変形手段6に制御を渡す。

【0026】つぎにループ構造変形手段6はループ構造解析手段3で生成された第1中間テキスト10およびル

8

ープ情報テーブル11と、ループ構造選択手段5で生成されたループ変形情報テーブル13とを入力し、初期値init、終値term、増分値incr×nの外側ループと、初期値i、終値MIN(i+(n-1)×incr, term)、増分値incrの内側ループとからなる2重ループ構造に変形した第2中間テキスト14を生成する。ここでiは外側ループの指標変数、MIN(n, m)はn, mのうち小さい値をもつ方を返す関数である。

【0027】たとえば、図2に示したループ構造は、初期値が1、終値が100、増分値が1、nが10なので、図3に示すように、外側ループの初期値が1、終値が100、増分値が10で、内側ループの初期値が外側ループの指標変数L、終値がMIN(L+(10-1)×1, 100)つまりMIN(L+9, 100)、増分値1であるような、2重ループ構造に変形される。この2重ループ構造の内側ループにおいては、条件④が満たされているためにベクトル化が可能となっている。

【0028】つぎに、ベクトル化手段7では、ループ構造変形手段6で変形された第2中間テキスト14を入力し、2重ループの内側ループ構造をベクトル処理の第3中間テキスト15に変形する。

【0029】最後に、命令生成手段8はベクトル化手段7で生成された第3中間テキスト15を入力して一連のベクトル命令およびスカラ命令を生成し、目的プログラム9を出力する。

【0030】図4を参照すると、本発明のベクトル化方式の別の実施例を適用したコンパイラ42の一実施例は、ループ構造解析手段43と、ループ構造選択手段44と、ループ構造変形手段45と、ベクトル化手段46と、命令生成手段47とから構成され、高級言語で書かれた原始プログラム41を入力し、ベクトル計算機に対する目的プログラム48を生成する。

【0031】なお、図4において、49、52、53、50、51はそれぞれコンパイラ42が処理の過程で生成する第1、第2、第3中間テキスト、ループ情報テーブル、ループ変形情報テーブルをあらわす。

【0032】また、目的プログラムを生成する対象となるベクトル計算機は、 $x(i) = x(i-1) + y(i)$ の形式の漸化式のベクトル命令を備えているものとする。

【0033】以下、本実施例の各部の機能および動作を説明する。

【0034】コンパイラ42は原始プログラム41を入力し、目的プログラム48を生成する。

【0035】このとき、ループ構造解析手段43は原始プログラム41中に現れるループ構造を認識し、そのループ構造に対する第1中間テキスト49を生成するとともに、そのループ構造の初期値init、終値term、増分値incrを格納したループ情報テーブル50

を生成する。

【0036】たとえば、図6のループ構造では、初期値3、終値12、増分値1というループ情報テーブル50が生成される。

【0037】つぎにループ構造選択手段44は、ループ構造解析手段43で生成された第1中間テキスト49およびループ情報テーブル50を入力し、ループ構造中に $x(i) = x(i-m) + y(i)$ の形の式が現れるかどうかを調べ、もしそのような式があったならば、 m がループ構造の増分値 $incr$ の倍数であるかどうかを調べる。そして、 m が $incr$ の倍数であればそのループ構造を変形によりベクトル化可能であるものとして、ループ変形情報テーブル51に m を格納し、つぎのループ構造変形手段45に制御を渡す。

【0038】たとえば、図6のループ構造では、代入文 d は $x(i) = x(i-n) + y(i)$ の形式に合致し、しかも $m=3=3 \times 1$ で、ループ構造の増分値1の倍数となっており条件が満たされているので、ループ変形情報テーブル51に3を格納する。

【0039】つぎにループ構造変形手段45は、ループ構造解析手段43で生成された第1中間テキスト49およびループ構造選択手段44で生成されたループ変形情報テーブル51を入力し、ループ構造を初期値0、終値 $m/incr-1$ 、増分値1で制御変数 i の外側ループと、初期値 $incr+i$ 、終値 $term$ 、増分値 n の内側ループとからなる2重ループ構造に変形した第2中間テキスト52を生成する。

【0040】たとえば、図6に示したループ構造は初期値3、終値12、増分値1、 $m=3$ であるから、図8に示すような、初期値0、終値 $3/1-1=2$ 、増分値1、制御変数 i の外側ループと、初期値が $1+L$ 、終値が12、増分値が3の内側ループからなる2重ループ構造に変形される。

【0041】つぎに、ベクトル化手段46では、ループ構造変形手段45で変形された第2中間テキスト52を入力し、2重ループの内側ループ構造をベクトル処理の第3中間テキスト53に変形する。

【0042】最後に、命令生成手段47はベクトル化手段46で生成された第3中間テキスト53を入力して一連のベクトル命令およびスカラ命令を生成し、目的プログラム48を出力する。

【0043】

【発明の効果】以上説明した本発明のベクトル化方式は、以下のような効果を有する。

【0044】後方依存関係のある間にまたがる定義参照の依存関係がある図2のようなループ構造もベクトル化できるため、実行速度がより高速な目的プログラムを生成できる。

【0045】 $x(i) = f(x(i-1))$ の形式の漸化式を処理するベクトル命令をもつベクトル計算機に対し、増分値 $incr$ のループ構造中に $x(i) = f(x(i-n))$ で n が $incr$ の倍数であるような形式の式が含まれる場合にもベクトル化できるようになり、より性能の高い目的プログラムを生成できる。

【図面の簡単な説明】

【図1】本発明の一実施例を適用したコンパイラの構成図である。

【図2】FORTRAN言語で記述された後方依存関係のあるループ構造の例を示す図である。

【図3】図2のループ構造を図1の実施例のベクトル化方式で変形したループ構造の説明図である。

【図4】本発明の別の実施例を適用したコンパイラの構成図である。

【図5】FORTRAN言語で記述された漸化式を含むループ構造の例を示す図である。

【図6】FORTRAN言語で記述された $x(i) = f(x(i-n))$ の形式の代入文を含むループ構造の例を示す図である。

【図7】図6のループ構造における配列Aの各要素の依存関係の説明図である。

【図8】図6に示したループ構造を図4の実施例のベクトル化方式で変形したループ構造の説明図である。

【符号の説明】

- 1、41…原始プログラム
- 2、42…コンパイラ
- 3、43…ループ構造解析手段
- 4…定義参照関係解析手段
- 5、44…ループ構造選択手段
- 6、45…ループ構造変形手段
- 7、46…ベクトル化手段
- 8、47…命令生成手段
- 9、48…目的プログラム
- 10、49…第1中間テキスト
- 11、50…ループ情報テーブル
- 12…定義参照テーブル
- 13、51…ループ変形情報テーブル
- 14、52…第2中間テキスト
- 15、53…第3中間テキスト

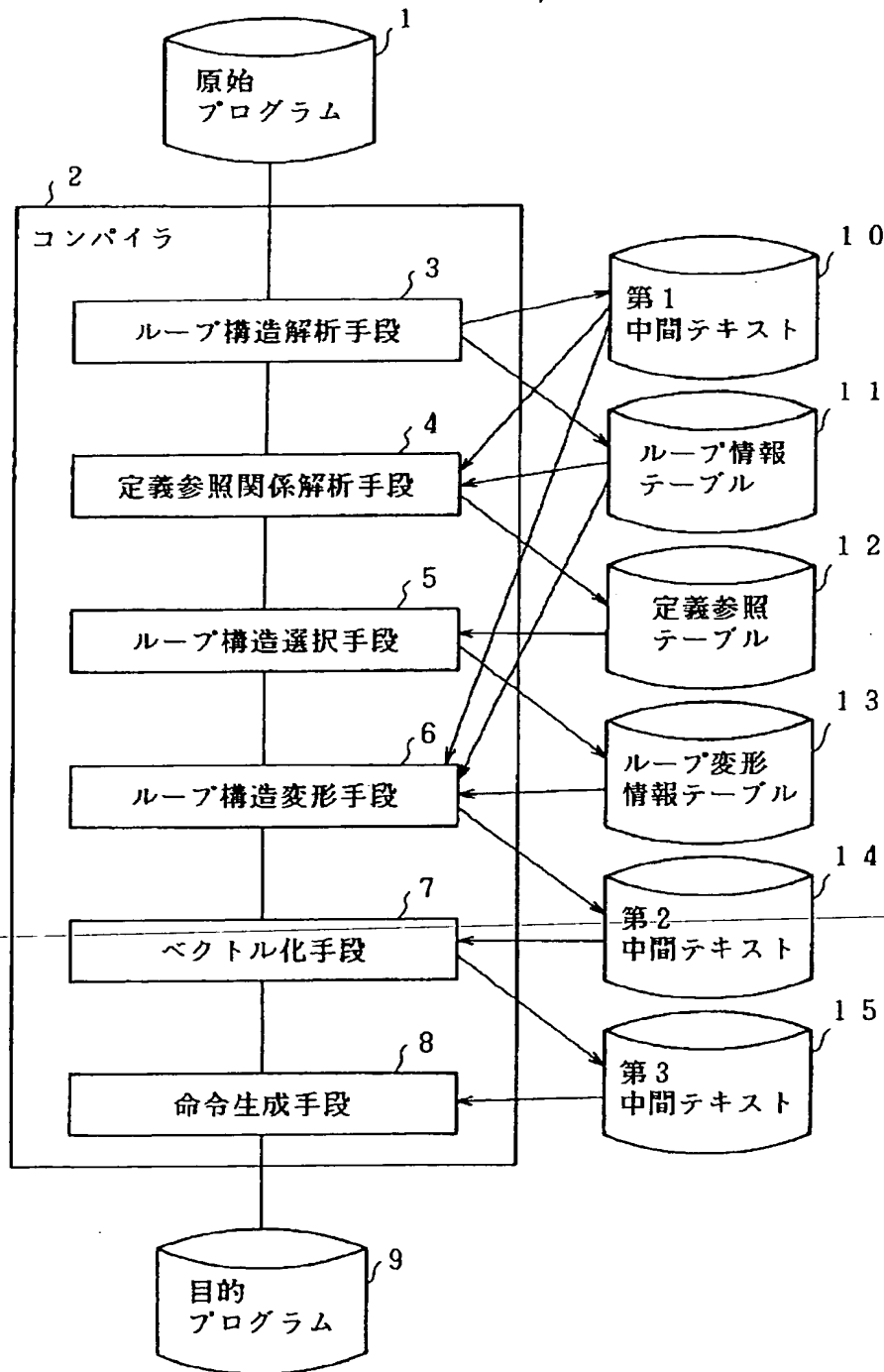
【図5】

```
DO 10 I=2, 10
10  A(I) = A(I-1) + B(I)      ..... c
```

【図6】

```
DO 10 I=3, 12
10  A(I) = A(I-3) + B(I)      ..... d
```

【図1】



【図8】

```

DO 10 L=0, 3-1
DO 10 I=1+L, 12, 3
10 A(I) = A(I-3) + B(I)

```

【図2】

```

DO 10 I=1, 100
  A(I) = B(I) * 2.0 ..... a
10 B(I+10) = A(I) + 1.0 ..... b

```

【図3】

```

DO 10 L=1, 100, 10
DO 10 I=L, MIN(L+9, 100)
  A(I) = B(I) * 2.0
10 B(I+10) = A(I) + 1.0

```

【図7】

(a) A

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1	4	7	10
---	---	---	----

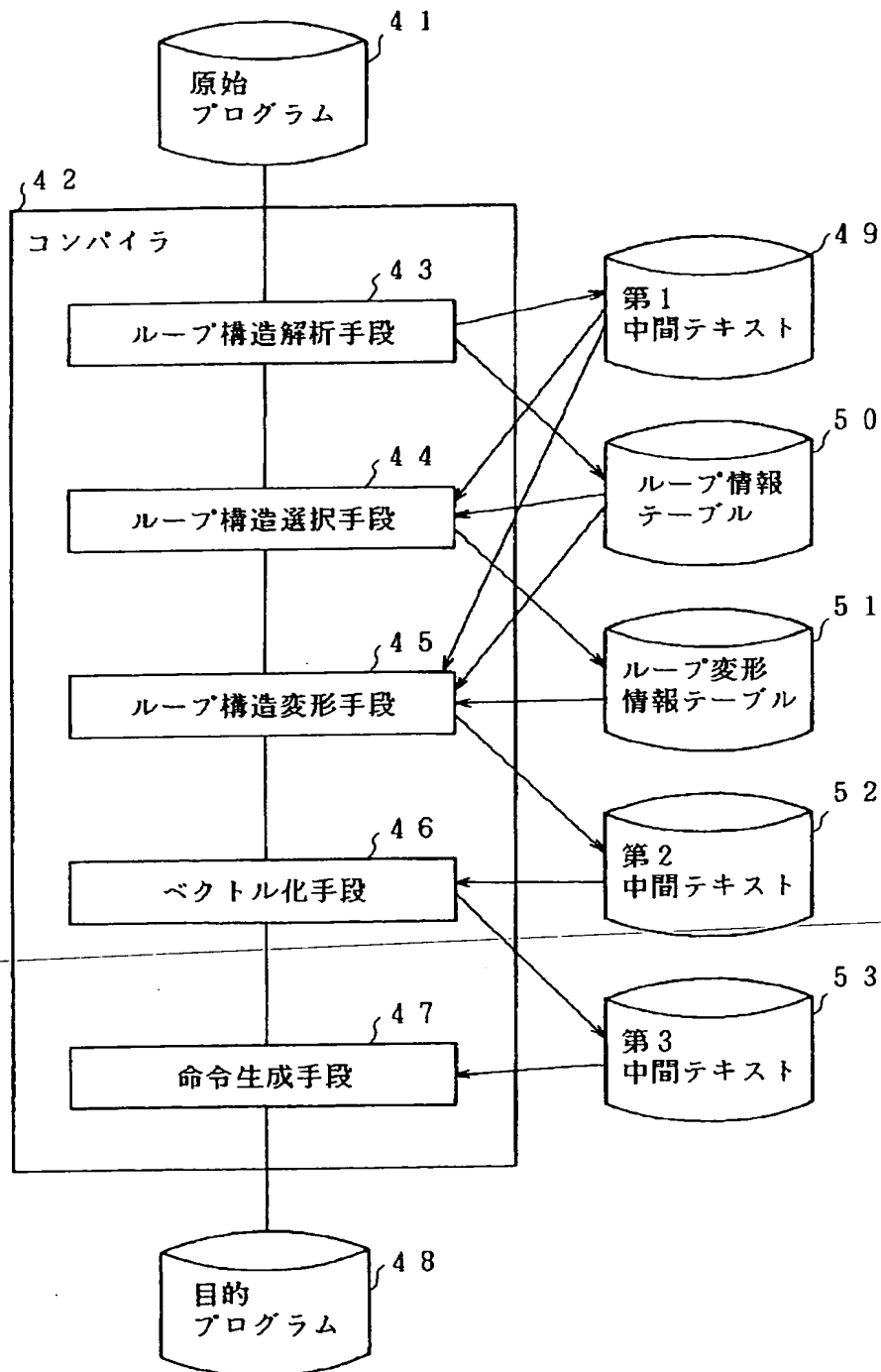
(b)

A

2	5	8	11
---	---	---	----

3	6	9	12
---	---	---	----

【図4】





THIS PAGE BLANK (USPTO)